

School Pick: Streamlining Student Dining and Meal Plans

Myunggyu Back, KangDeok Seo, JaeHong Lee, YeonSeo Choi

Sungkyunkwan University, 2066, Seobu-ro, Jangan-su, Suwon-si,
Gyeonggi-do, Republic of Korea

Abstract

The process of choosing a cafeteria for students and making meal plans with friends is often inconvenient. Therefore, we created the web application School Picks to provide convenience for students' dining decisions and meal appointments. This application reflects the user's selected menu and prices, providing intuitive information about restaurant recommendations and location information through Naver Maps. Additionally, it offers a personal page for sharing restaurant appointment information, delivering timely updates to individual users. We designed our solution with a user-centric menu recommendation system, focusing on effective UI/UX design. To convey convenience effectively, all designs are web-based but optimized for mobile use. Moreover, we concentrated on developing a simple web application that allows quick and efficient access to information through intuitive information delivery using APIs and utilizing a database for personal appointments and history.

Keywords: Restaurant information · university life · Meal mate · convenience service

1 Introduction

School Picks is a web application that alleviates students' concerns related to meals and facilitates easy meal appointments with friends.

We believe that the most common dilemmas students face regarding meals in university life can be categorized into two main issues. The first is the inconvenience of finding information about new restaurants, leading to a tendency to stick to familiar ones. The second is the cumbersome process of coordinating opinions on meal locations and menu choices when making meal appointments. Typically, people resort to platforms like Naver or Google to find information about restaurants during this process, but they often find themselves conflicted amidst numerous reviews and information. School Picks identifies the root cause of this problem as providing too many choices.

To address these issues, School Picks brainstormed various ways to deliver information about meals in a diverse and enjoyable manner. Moreover, to reduce existing complexities, efforts were made to eliminate unnecessary processes such as login, installation, and authentication, while retaining essential features to create a convenient web application.

Our first goal was to simplify the process of choosing a restaurant, reducing the complexity of choices and making the experience enjoyable. To achieve this, we opted for a random selection method and made efforts to convey convenience and essential information accurately. Additionally, in the recommendation feature, we restricted the data range to the campus of Sungkyunkwan University in the fields of natural sciences and humanities for practical implementation.

Our second goal was to eliminate ambiguity in the process of dining with friends and devise a definite method for making appointments. In line with the culture of sharing on social media, we created a dining companion-finding page that allows collaborative reservation-making. The objective of this page is to enable easy page creation, enforce necessary information entry, provide information verification, and ensure privacy protection.

All designs of School Picks were based on the assumption that it is in the initial stage of gathering users. Therefore, although the service is simple and convenient for users, it was designed with scalability in mind. For convenience, the frontend was developed using Thymeleaf-based design tailored for mobile devices. Additionally, for reliability, the backend framework Spring and MySQL as the database were chosen, separating the development stages of service, DB design, and SQL writing.

2 Design for service

2.1 MVC Pattern Architecture

The School Picks team utilized the MVC pattern while working on a Spring-based web project. The MVC pattern stands for Model-View-Controller and is a widely used software design pattern for implementing user interfaces, data, and logic control.

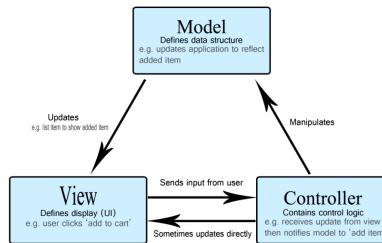


Figure 1: MVC Pattern Architecture

The advantage of employing this pattern is the separation of concerns, dis-

tinguishing between the software’s business logic and its user interface. This “separation of concerns” facilitates clearer division of tasks among team members during collaboration, enabling them to specialize in their respective areas and enhance overall management. By designing the system using the MVC pattern, our team anticipated that, during collaboration, team members could work on different tasks without impacting each other. This was expected to result in faster implementation.

Utilizing the MVC pattern, School Picks team members collaborated by assigning four individuals to handle data collection, data processing, UI design, and interface design, respectively.

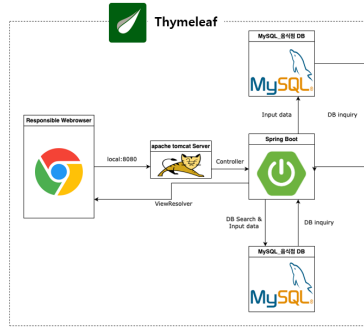


Figure 2: System Architecture Overview

The diagram illustrates our overall system architecture designed according to the MVC pattern. Apart from HTML and CSS, the frontend code was written in JavaScript, and the backend code was implemented in Java. To manage both frontend and backend on a single server, we used the Thymeleaf library. This allowed for consistent code sharing among team members and facilitated testing individual modifications immediately in their local environments through data injection code.

2.2 Related Tool

In this project, GitHub and Figma were utilized for collaboration among team members. GitHub was employed for code management, while Figma was used to share UI/UX designs. The locofy.ai tool was employed to convert Figma designs into web pages for practical implementation.

Thymeleaf, chosen for its ability to operate in both web and non-web environments, is a Java XML/XHTML/HTML5 template engine, making it suitable for the project. Additionally, MySQL, a freely available RDBMS, was selected for the database. Spring Data JPA was employed as a tool to facilitate query writing when retrieving data from the database.

Furthermore, to enhance the utility of the meal recommendation feature with map information and sharing with friends, Naver Map API and Kakao

Share API were integrated into the project.

2.3 Challenges

The most challenging aspect of the School Picks project was finding a balance between the goal of being easily accessible and user-friendly and the practical issues encountered during implementation and usage. I had removed the login feature, assuming that not many people would install an application or sign up just for a simple meal. However, this decision led to several problems.

2.3.1 Inability to Confirm Friends in the “Find Mate” Feature

The absence of user registration posed a challenge in identifying friends when they left their names for a meal, making it difficult to distinguish individuals. However, reintroducing user registration was deemed inconvenient for the sake of user-friendliness. Therefore, when users create a page, they are requested to leave a specific message that the author can use for identification. Additionally, a verification process was implemented for friends leaving their names, ensuring that the author could confirm whether the name was entered accurately.

2.3.2 Lack of Access to Past History Information

To enhance the user experience and streamline the process for users who wish to have another meal using the application, I thought it would be more efficient for them to select friends to send a message based on their past records. Therefore, I created a History page, allowing users to search for past posts using the author and title. However, I considered the possibility of users forgetting the same author name and password, so to prevent this, I implemented cookies in the browser. This way, if a user creates a new page within a week, the previous record is automatically entered.

2.3.3 Privacy and Personal Information Protection

Concerns arose regarding potential privacy breaches when users shared their contact information or real names for communication purposes. To address this, an anonymous feature was introduced during page creation, accessible through the author’s modification password. This way, even if the server is compromised, the password cannot be accessed without internal processing.

3 Implementation

3.1 Database

We created two tables in the database for the School Picks project: the “Shop” table for the selection system and the “FindMateRoom” and “RoomUser” tables for the meal appointment system.

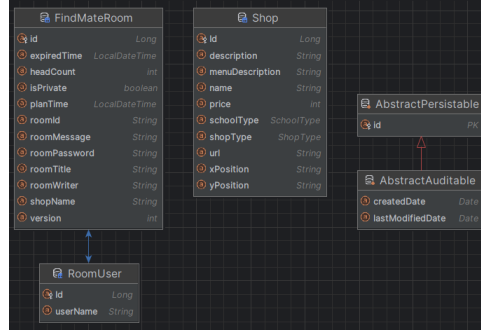


Figure 3: Entity-relationship Diagram for School Picks.

In the “Shop” table, information such as coordinates, descriptions, and search conditions is stored. Since the purpose is limited to retrieving information for service-related queries, a single table was used.

For the meal appointment system, we needed to manage details such as the time required for scheduling appointments and additional information for new participants in the appointment process. To handle this dynamic situation, we divided the information into two tables: “FindMateRoom” and “RoomUser.”

3.2 Menu Recommendation System using Claw machine

The selection system allows users to quickly choose a restaurant by selecting campus, type of cuisine, and price, resulting in a recommendation for one restaurant. The goal of this system is to provide convenient, enjoyable, and useful information when selecting a restaurant. To achieve this, the UI/UX was designed in a mobile format. The aim was to make it immediately usable upon entering the webpage and to enhance the fun of discovering new restaurants by considering options beyond the usual places students frequent.

3.2.1 UI/UX

The selection system allows users to quickly choose a restaurant by selecting campus, type of cuisine, and price, resulting in a recommendation for one restaurant. The goal of this system is to provide convenient, enjoyable, and useful information when selecting a restaurant. To achieve this, the UI/UX was designed in a mobile format. The aim was to make it immediately usable upon entering the webpage and to enhance the fun of discovering new restaurants by considering options beyond the usual places students frequent.

3.2.2 Data/Back-end Service

In the recommendation system, the area where we exerted the most effort, aside from the UI, was in data collection. While there are methods such as web

scraping from platforms like Naver or Google to gather information related to restaurants, this approach comes with two main issues.

Firstly, there is a lack of clarity in the collected data. Based on personal experience in researching data, approximately 20percent of restaurants did not list their menu items in text but rather presented them as images. Additionally, some uploaded outdated price lists without reflecting the current prices.

Secondly, the collected data often lacks conciseness. For restaurants that one hasn't visited, it can be cumbersome to make menu choices without being able to rely on reviews.

To address these challenges, we sought information from blogs to confirm unlisted prices and, for familiar restaurants, selected the two most popular dishes based on reviews if the specific dishes weren't known. Furthermore, in the database, we calculated and stored the average estimated cost per person based on the prices and servings of these selected dishes. This approach aimed to enhance the convenience of the system, aligning with the goal of providing useful and practical information to users.

shop_id	description	menu_description	name	price	school_type	shop_type
179	저렴한 가격의 고깃집, 단체 회식으로도 좋다. ...	삼겹살 500g 25000원 생...	가톨릭고기	12500	NSC	RESTAUR...
180	탈장살이 맛있다고 한다. 가격이 좀 있다.	제주삼겹살 19000원 제주...	제주도야자...	20000	NSC	RESTAUR...
181	교수님 카드, 특별한 선배 있으면 뽐 가는 식당, ...	임대지삼겹살 600g 470...	삼다육장	20000	NSC	RESTAUR...
182	고깃집 중 특히 좋아하는 삼다육장 다들이다. 가격도...	주먹고기 15000원 삼겹살 ...	산다라주먹...	15000	NSC	RESTAUR...
183	좋은 고기를 저렴한 가격에 먹을 수 있는 곳. 서...	주먹고기 200g 18000원 ...	보리내주먹...	15000	NSC	RESTAUR...
184	왕복음밥, 육개장만 먹으려 한다. 일별이 먹한듯...	왕복음밥 7000원 육개장밥...	애국밥	7000	NSC	RESTAUR...
185	로또만 이겨는 빠는 것만 탈하고 소스는 추천소...	에그마요 5500원 주키세트...	서보왕이	7000	NSC	RESTAUR...
186	맛집에 비해 가성비가 많이 나빠졌지만, 여전히...	돈가스 7500원 매월정식 소...	분북돈가스	8000	NSC	RESTAUR...
187	신입생 때 돈가스 살면 조금이 아껴보지만 나중...	돈살면 9500원 돈가스 2점...	돈살	9000	NSC	RESTAUR...
188	맛나 술 많이 먹고 재밌게 가는 곳, 밥 먹으려...	백해장국 9000원 신지해장...	일인자갈밥	9000	NSC	RESTAUR...
189	자주 안 가겠지만 일한지라는 다른 술은 감자탕...	백해장국 9000원 신지해장...	서울24시...	9000	NSC	RESTAUR...
190	저렴한 가격에 새다른 기분들 날 수 있다. *자주...	자주된 붕가게우동 8000원...	카리타게...	8000	NSC	RESTAUR...
191	왕복과 왕복에 사후 매점과 다들 먹어보았...	왕복국수 4000원 마늘된장...	왕복에 후미	4000	NSC	RESTAUR...

Figure 4: Example of DB

Additionally, we explored the application of models for aggregating reviews or analyzing reviews. However, due to the difficulty in determining the authenticity of reviews, particularly in distinguishing promotional reviews, we opted to collect data by aggregating reviews that could be objectively assessed through firsthand information and opinions from posts on platforms like Everytime.

In this manner, we collected information on a total of 222 establishments, saving their summarized details, coordinates, and the URL of the search results page.

On the backend, we utilized Spring Data JPA to dynamically generate SQL queries based on user-set conditions, enabling the creation of dynamic queries. This allowed users to dynamically set conditions on the web page through Thymeleaf, and upon receiving these conditions, the backend retrieved the data and returned it to the web page.

3.3 Find Mate using meal appointment system

The meal appointment system in School Picks is a program that generates a shareable page URL when essential values for a meal appointment with friends are entered. Users who receive the link can either directly send the URL to their friends or use the KakaoTalk or Instagram share buttons to transmit it.

When friends receive the message, they can input their information, such as their name or contact details, based on the message left by the person who made the reservation, indicating their interest in dining together.

The key advantage of this feature is that, unlike the previous method of proposing appointments to a specific few through KakaoTalk group invites or one-on-one chats, sharing the URL allows for proposals to a larger and unspecified audience. Anticipated benefits include reducing the time spent contacting friends available at the same time, thereby expediting meal matching and saving time. Additionally, it is expected to eliminate ambiguity in appointments. When making plans with friends through KakaoTalk, a common issue arises when the choice of restaurant and the timing of the meal are unclear, especially when many friends are involved, making decisions more challenging. By confirming the restaurant and time in advance and providing a feature to search for information about past meals with friends, this system aims to facilitate finding meal companions quickly and eliminate uncertainties in appointment planning.

3.3.1 UI/UX

For the page where users make appointments, we included essential fields such as the restaurant name, date and time, number of people, and a message. Additionally, we introduced several other items to enhance functionality.

3.3.2 Choosing a Concept

Recognizing the need for users to share SchoolPicks images directly on social media, it was crucial that the provided images align with the user's preferences. To address this, we designed three concepts, allowing users to manually select from these options or have them randomly assigned during the appointment creation process. The images for each of the three concepts are as follows.



Figure 5: Concept of appointment creation page

3.3.3 Friend Name Visibility

As the link is shared on social media platforms where it can be viewed by a diverse audience, there may be individuals who wish to dine together but

prefer not to expose their personal information. Additionally, in cases where the participants are not friends, leaving contact details might be necessary for communication purposes.

To address these concerns, we created a private mode. In private mode, even if someone enters their name to express interest in the meal appointment on the page, individuals other than the creator of the appointment cannot view that name. To access the names of friends, one must input the password set during the creation of the appointment.

3.3.4 SNS Sharing System

Once the appointment is complete, School Picks provides a URL for sharing, and there are two methods for sharing this URL. The first method involves copying it to the clipboard and pasting it wherever desired, providing higher flexibility. However, in shared spaces accessible to everyone, additional information beyond just the name may be necessary for identification. The second method involves utilizing Instagram or Kakao API for sharing. In a mobile environment where the user is already logged into these social media platforms, sharing with friends becomes quicker. When sharing via KakaoTalk, details such as the restaurant, number of people, appointment time, and the deadline will be included in the shared information.

3.3.5 Post Retrieval System

The meal appointment system operates on a simple premise: users create and share pages easily, and the recipients of the page link input their names. Since all processes occur without requiring login, there was no way to gather information about individuals with whom users had shared meals in the past. To address this, School Picks introduced a History page, enabling users to retrieve past posts by searching using the author's name or the post title.

4 Data/Back-end Service

In the meal appointment system, we implemented CRUD functionality for pages, allowing users to Create, Read, Update, Join, and Delete pages. Each page is associated with service implementation requests to facilitate the corresponding functionalities.

4.1 Page Creation Service

When creating a post, we designed the system to transform the basic information such as the desired number of people, deadline, message, etc., received from the web page into the appropriate format for the MySQL table. Subsequently, this transformed data is stored in the database. During this process, we implemented rules for rejecting requests based on the following criteria for the appointment time and deadline.

Appointment Time: The appointment time is determined based on the server's reception time, specifically 30 minutes after the received time. If the proposed appointment time is less than 30 minutes from the current time, the system judges that it is not sufficient time to arrange a meal appointment and sends a rejection message.

Deadline Time: The deadline time is the user-configured time when they no longer wish to receive requests before the appointment time. If this time is not set between the current time and the appointment time, the system sends a rejection message.

4.2 Page Retrieval Service

Public Mode: In public mode, excluding the password, all information, including details provided by individuals expressing interest in sharing a meal, is accessible. In this scenario, verification is not required for the password, allowing unrestricted access to all the information.

Private Mode: In private mode, it operates similarly to public mode, but the information provided by individuals expressing interest in sharing a meal is concealed and not visible during regular viewing. In this case, upon receiving the password from the web page, if it matches the one associated with that particular page, the user's information is then revealed.

4.3 Page Modification Service

In the meal appointment system, users should have the ability to modify the information if they made an error in input or wish to reschedule the appointment. To enable this, the system prompts the user to enter the password associated with the page. If the entered password matches the one saved for the page, the system navigates the user to a page with the same UI as during the initial creation.

On this page, the user can make modifications to the information as needed. Any changes made are then reflected in the database, ensuring that the updated details are saved and accurate.

4.4 Page Deletion Service Service

If a user no longer wishes to retain their information on the server or if an appointment made through a created and shared page doesn't unfold as desired, the user may want to delete the post. In this case, pressing the delete button on the respective page allows the user to delete the page and the user friend information within it, using the password and roomId associated with the creation of the page.

4.5 Page Join Service Service

If someone who receives the shared page wishes to participate, they can enter their name or contact information into the input field on the page. Upon

pressing the button, the contents of the input field, along with the RoomId, are stored in the database.

5 Evaluation

Service	Test Coverage	Result
Restaurant recommendation	Do program generate appropriate queries based on input conditions during the drawing process?	P
	Do program retrieve the correct data in the drawing process for the Detail?	P
	In the drawing process, does it retrieve the predetermined result value when there is no target?	P
	During the drawing process, does it bring different results based on the order even when the same value is input?	P
Find Mate Page	Is the creation, modification, retrieval, and deletion of rooms occurring successfully within the database?	P
	Is error handling occurring properly in situations such as exceeding the deadline, surpassing the set number of participants, etc.?	P
	Is the addition of users being applied correctly to the database?	P
	Does the room user query provide the desired data when it is public or private?	P
	Does the process of querying History deliver data that matches the search criteria?	P

Figure 6: Backend Testing Criteria

Due to delays in implementation, we did not receive bug reports from actual users. Therefore, for backend testing, we utilized JUnit5, a testing framework commonly used in test-driven development with the Java programming language. On the frontend side, the evaluation was conducted not at the code level but by directly executing the program.

This approach allowed us to assess the functionality and identify any potential issues by employing automated tests for the backend using JUnit5, while the frontend evaluation involved manual testing by running the program itself.

Service	Test Coverage	Result
Restaurant recommendation	Do the buttons work as intended?	P
	Are page transitions and HTTP requests functioning as intended?	P
	Are there inputs that cause errors?	P
	Is there a case where the detail information goes beyond the screen?	P
	Is the Naver Maps API functioning properly?	P
Find Mate Page	Do the buttons work as intended?	P
	Are page transitions and HTTP requests functioning as intended?	P
	Are there inputs that cause errors?	P
	During the process of creating a page, does it display appropriate messages in case of error situations?	P
	Is information about the configured page appropriately retrieved during page retrieval?	P
	Is data output in public mode and private mode coming out as intended?	P
	Is the Instagram or KakaoTalk API operating as intended?	P

Figure 7: Frontend Testing Criteria

While inspecting these aspects, we identified potential errors that could occur during the process of fetching data and making requests after entering values

on the screen. In response, we implemented a mechanism to determine whether the request on the server side was successful or failed. Depending on the situation, appropriate error messages were generated and sent, and this information was then displayed in popup windows on the frontend. This approach ensures that the system functions as intended, providing users with meaningful error feedback.

6 Limitation

6.1 Lack of Convenience Features

During the page creation process, considering the idea that it would be beneficial to send mobile alerts via SMS or KakaoTalk when friends accept the meal invitation on the web, it was found that SMS messages cost 20 won per message, and for KakaoTalk messages, a separate contract for sending notifications was required, costing 8 won per message. As SchoolPicks is a non-profit service, this option was not chosen. Additionally, while email is free, it was not considered optimal for accessibility, leading to the adoption of a deadline-setting approach.

Therefore, while real-time alerts for each friend accepting the meal invitation might not be feasible, checking the page after the deadline allows users to see all accepted participants in one go. This approach helps mitigate the issue of needing to continuously verify whether friends have entered their names.

6.2 Difficulty in Maintaining and Expanding the Service

The data input into the recommendation system’s database, obtained through personal research rather than web scraping, is believed to be inconsistent with the direction the service is aiming for. This can pose challenges when expanding the service to universities other than Sungkyunkwan University. Moreover, if the data is not regularly updated, it may lead to a decline in the quality of the service over time.

The most effective solution to this issue is considered to be a combination of web scraping for certain accessible establishments and incorporating a user-driven registration and modification request system. By allowing users to submit requests for new additions or modifications within School Picks, the service can quickly collect and update data, ensuring a more dynamic and responsive approach to changes.

7 Related Work

7.1 Naver Maps and Google Maps

Naver Maps and Google Maps are commonly used platforms to obtain information about restaurants, offering extensive details on menus and reviews that can be quite useful. However, I believe that the abundance of information can also introduce a drawback by making the decision-making process more

challenging. In School Picks, random recommendations are implemented to streamline the restaurant selection process, aiming to reduce the time spent on deciding where to eat.

7.2 Skkuchin

Skkuchin is a platform designed to help Sungkyunkwan University students find friends to dine with. While it offers features such as finding friends, a restaurant “World Cup” and restaurant recommendations, the friend-finding function faces challenges due to a low number of regular users, resulting in infrequent successful matches. Additionally, the restaurant recommendation feature lacks a sufficient pool of options, leading to repeated recommendations of the same restaurants.

In response, School Picks aimed to address the need for quickly choosing a place to eat and finding dining companions seamlessly, even when the user pool might be limited. The result is a recommendation system through random selection and a service for finding dining companions through sharing on social media. The recommendation system narrows down choices and provides a concise evaluation in one line, aiding users in making quick decisions. The dining companion service allows users to share a link without the need for logins or cumbersome procedures, facilitating easy coordination and eliminating unnecessary adjustments during the appointment process.

8 Conclusion

School Picks has focused on quickly helping students find restaurants they want to eat at and efficiently gathering friends to dine together without conflicts. To achieve the goal of speed and simplicity, a web application was chosen, and complex procedures such as logins were eliminated. However, to address potential issues arising from this choice, various features like History and an anonymous system were introduced to prevent possible problems.

In the restaurant recommendation system, data was processed to present information in an easily understandable format. In the dining companion-finding feature, efforts were made to automate the input process to align with the goal of achieving maximum speed. These initiatives aimed to provide a convenient experience for students who may have felt inconvenience in the process of selecting a restaurant or arranging meal appointments, offering them a more convenient experience compared to existing applications.